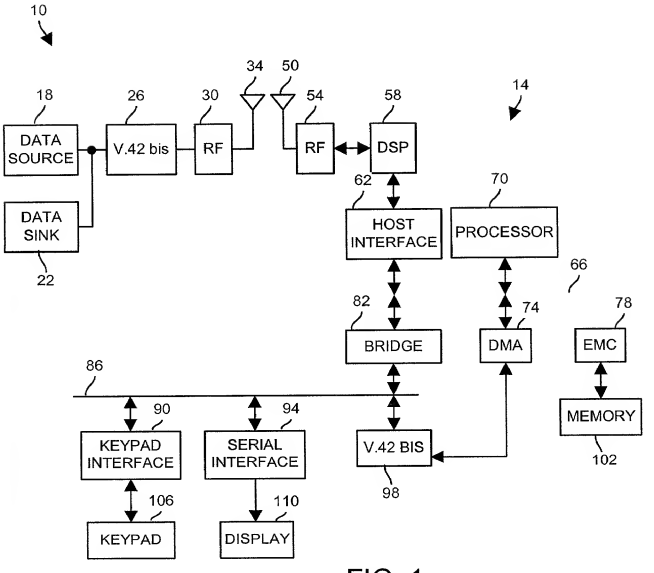

10
14
18
DATA SOURCE
26
V.42 bis
30
RF
34
50
54
RF
58
DSP
62
HOST INTERFACE
70
PROCESSOR
66
DATA SINK
22
82
BRIDGE
74
DMA
78
EMC
MEMORY
102
86
90
KEYPAD INTERFACE
106
KEYPAD
94
SERIAL INTERFACE
110
DISPLAY
V.42 BIS
98

FIG. 1

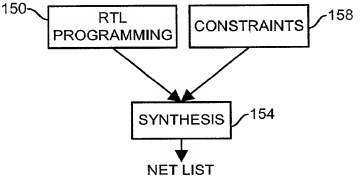

150
RTL PROGRAMMING
CONSTRAINTS
158
SYNTHESIS
154
NET LIST

FIG. 2

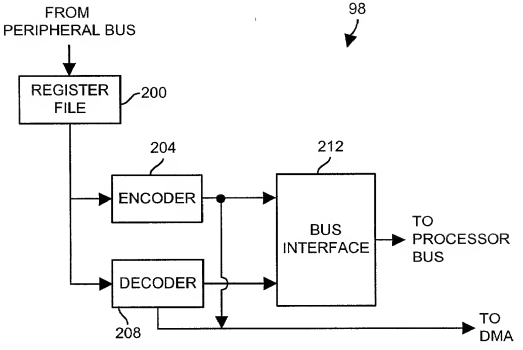

FROM
PERIPHERAL BUS
98
REGISTER
FILE
200
204
212
ENCODER
BUS
INTERFACE
TO
PROCESSOR
BUS
DECODER
208
TO
DMA

FIG. 3

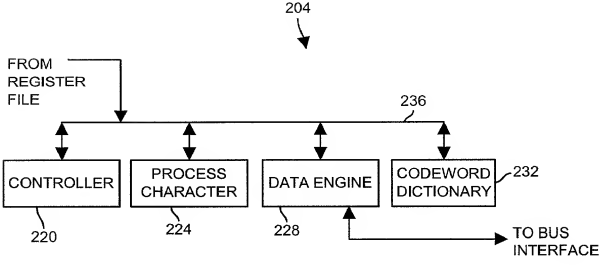

204
FROM
REGISTER
FILE
236
CONTROLLER
PROCESS
CHARACTER
DATA ENGINE
CODEWORD
DICTIONARY
232
220
224
228
TO BUS
INTERFACE

FIG. 4

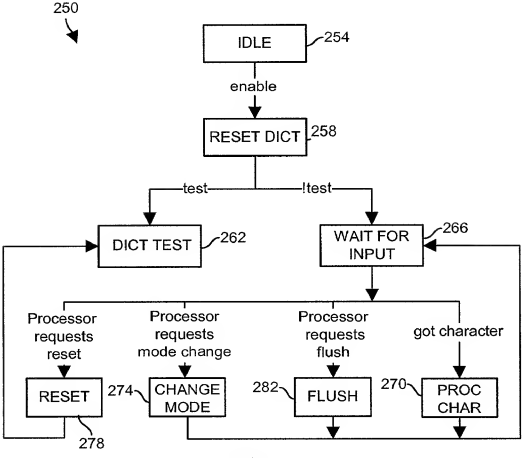

250
IDLE
254
enable
RESET DICT
258
test
!test
DICT TEST
262
WAIT FOR INPUT
266
Processor requests reset
Processor requests mode change
Processor requests flush
got character
RESET
274
CHANGE MODE
282
FLUSH
270
PROC CHAR
278

FIG. 5

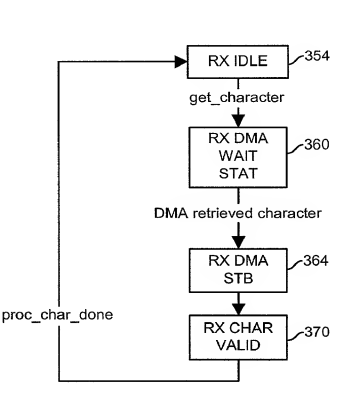

350
RX IDLE
354
get_character
RX DMA WAIT STAT
360
DMA retrieved character
RX DMA STB
364
proc_char_done
RX CHAR VALID
370

FIG. 7

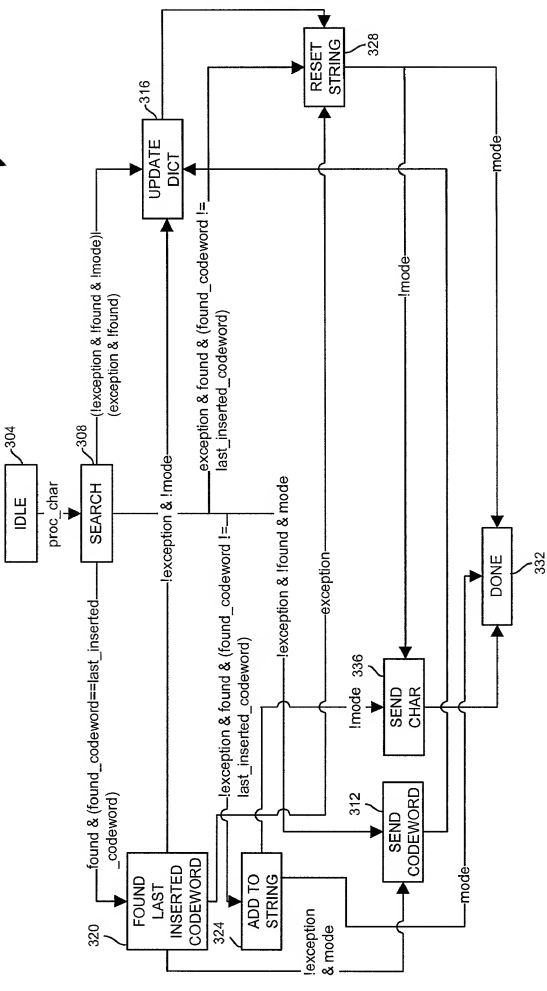
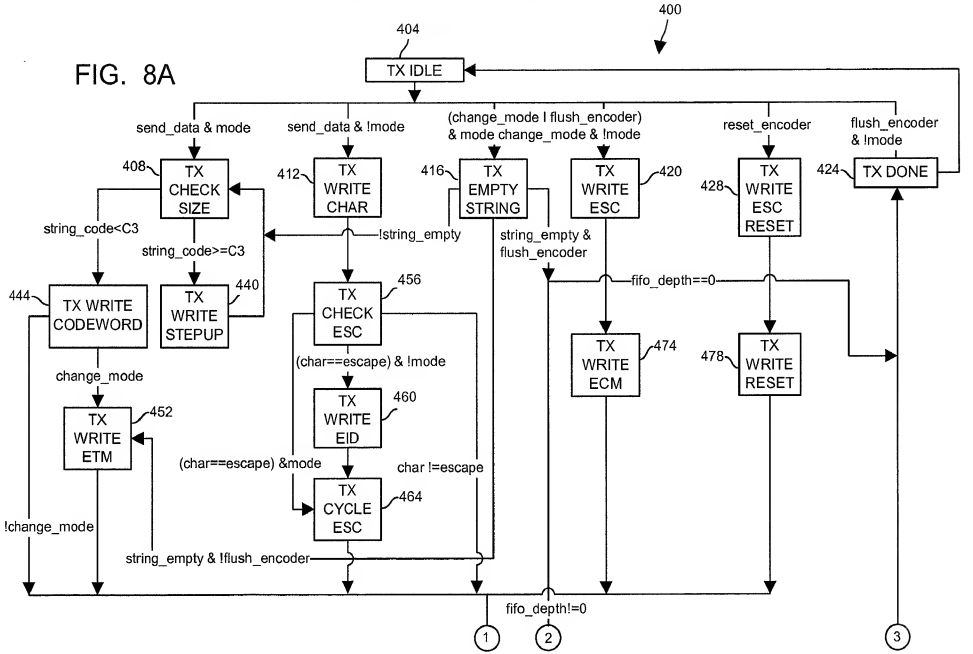

300
IDLE
304
proc_char
SEARCH
308
(!exception & !found & !mode)||
(exception & !found)
UPDATE
DICT
316
RESET
STRING
328
found & (found_codeword==last_inserted
_codeword)
!exception & !mode
FOUND
LAST
INSERTED
CODEWORD
320
!exception & found & (found_codeword !=
last_inserted_codeword)
ADD TO
STRING
324
!exception & !found & mode
exception & found & (found_codeword !=
last_inserted_codeword)
!exception & !mode
SEND
CODEWORD
312
exception
!mode
SEND
CHAR
336
!mode
mode
mode
DONE
332

FIG. 6

TX2280 TX7160

FIG. 8A



Docket No.: 29341/PC0058
Title: Data Compression System
Inventors: Matthews, Phillip M.
Sheet 5 of 18
(fig. 8A)

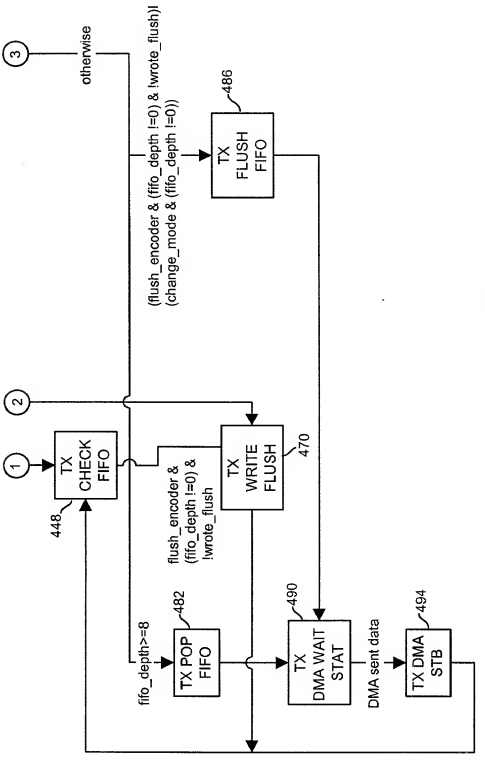

1
2
3
otherwise
(flush_encoder & (fifo_depth !=0) & !wrote_flush)|
(change_mode & (fifo_depth !=0))
TX FLUSH FIFO
486
TX CHECK FIFO
448
flush_encoder & (fifo_depth !=0) & !wrote_flush
TX WRITE FLUSH
470
fifo_ depth>=8
TX POP FIFO
482
TX DMA WAIT STAT
490
DMA sent data
TX DMA STB
494

FIG. 8B

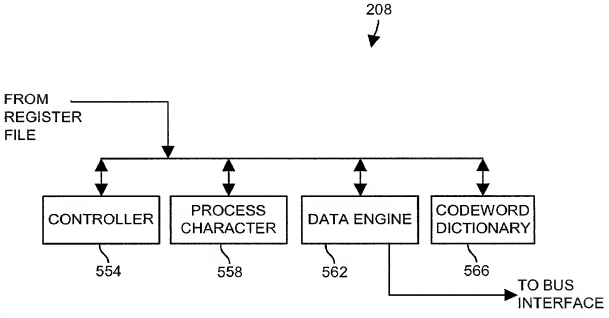

208
FROM REGISTER FILE
CONTROLLER
PROCESS CHARACTER
DATA ENGINE
CODEWORD DICTIONARY
554
558
562
566
TO BUS INTERFACE

FIG. 9

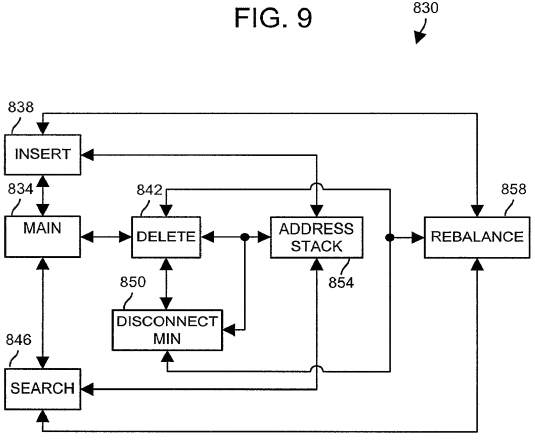

830
838
INSERT
834
MAIN
842
DELETE
ADDRESS STACK
858
REBALANCE
850
854
DISCONNECT MIN
846
SEARCH

FIG. 14

TO2200 FOTTH660

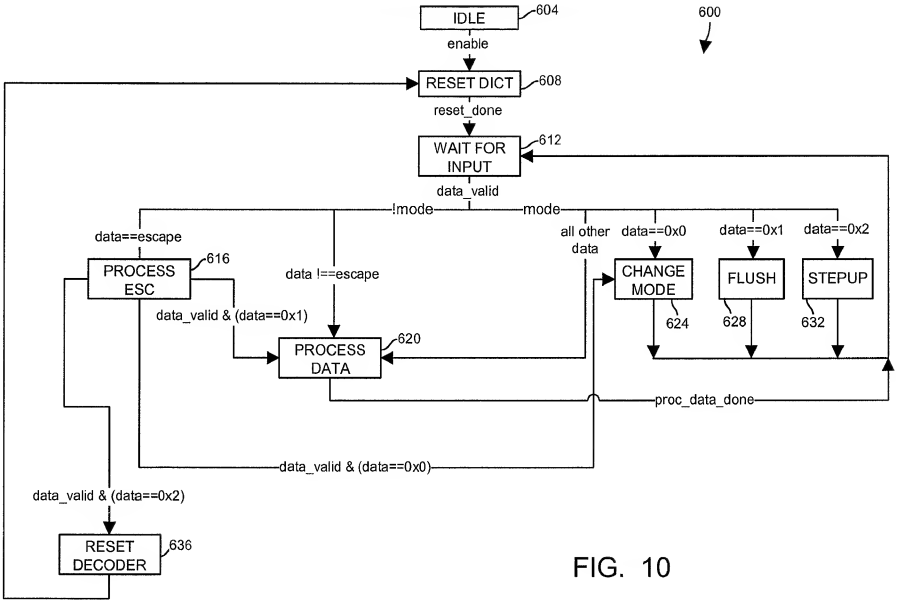


FIG. 10

Docket No.: 29341/PC0058
Title: Data Compression System
Inventors: Matthews, Phillip M.
Sheet 8 of 18 (fig. 10)

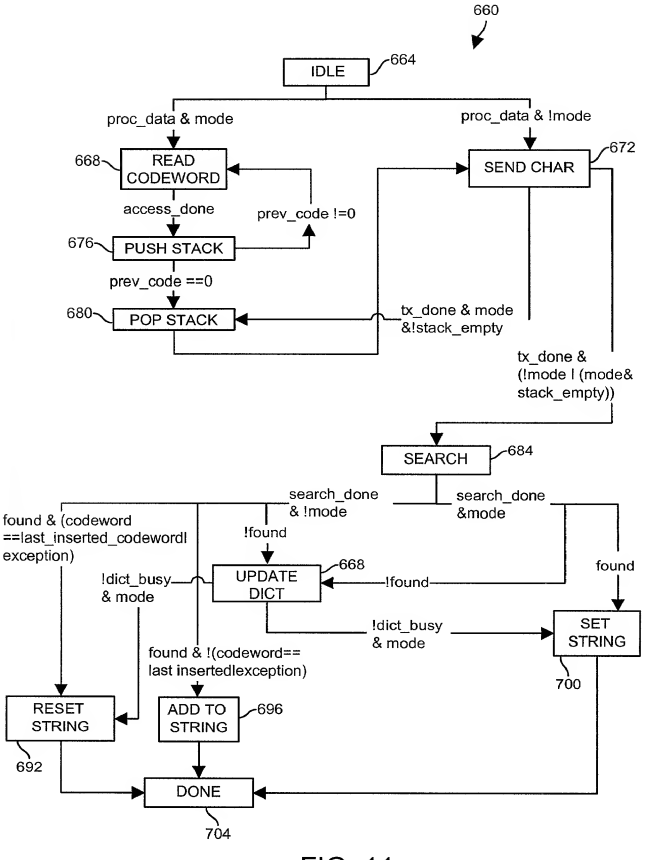

660
IDLE
664
proc_data & mode
proc_data & !mode
668
READ CODEWORD
SEND CHAR
672
access_done
prev_code !=0
676
PUSH STACK
prev_code ==0
680
POP STACK
tx_done & mode &!stack_empty
tx_done & (!mode | (mode& stack_empty))
SEARCH
684
search_done & !mode
search_done &mode
found & (codeword ==last_inserted_codeword| exception)
!found
found
!dict_busy & mode
UPDATE DICT
668
!found
!dict_busy & mode
SET STRING
700
found & !(codeword== last insertedlexception)
RESET STRING
692
ADD TO STRING
696
DONE
704

FIG. 11

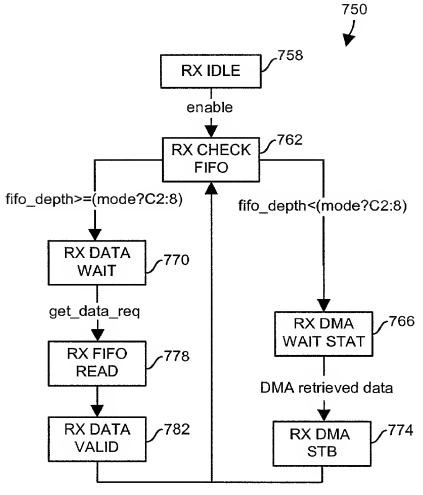

750
RX IDLE
758
enable
RX CHECK FIFO
762
fifo_depth>=(mode?C2:8)
fifo_depth<(mode?C2:8)
RX DATA WAIT
770
get_data_req
RX FIFO READ
778
RX DATA VALID
782
RX DMA WAIT STAT
766
DMA retrieved data
RX DMA STB
774

FIG. 12

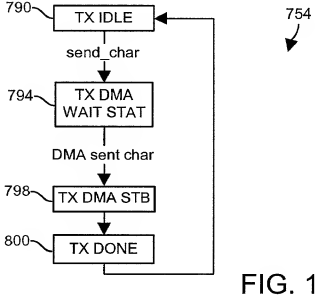

790
TX IDLE
754
send_char
794
TX DMA WAIT STAT
DMA sent char
798
TX DMA STB
800
TX DONE

FIG. 13

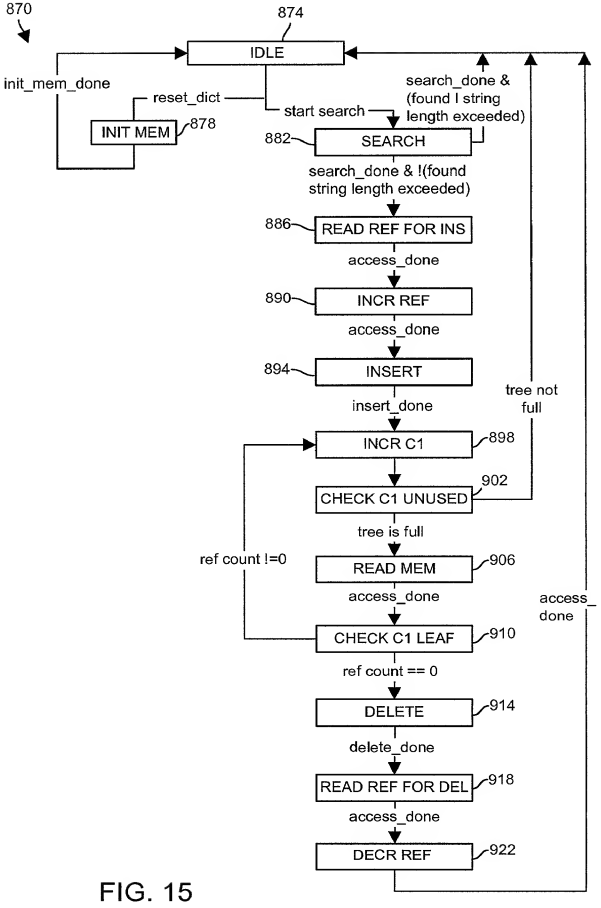

870
874
IDLE
init_mem_done
reset_dict
start search
search_done & (found I string length exceeded)
INIT MEM
878
882
SEARCH
search_done & !(found string length exceeded)
886
READ REF FOR INS
access_done
890
INCR REF
access_done
894
INSERT
insert_done
tree not full
INCR C1
898
902
CHECK C1 UNUSED
tree is full
ref count !=0
READ MEM
906
access_done
access_done
CHECK C1 LEAF
910
ref count == 0
DELETE
914
delete_done
READ REF FOR DEL
918
access_done
DECR REF
922

FIG. 15

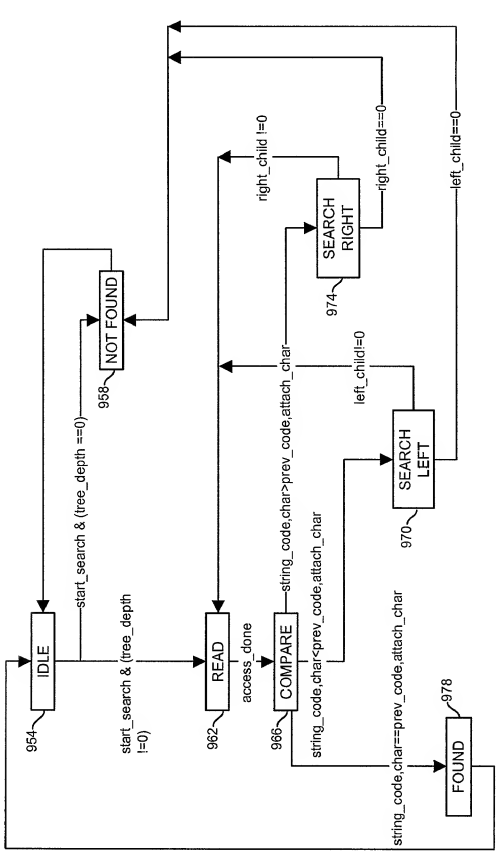

950
IDLE
954
start_search & (tree_depth ==0)
NOT FOUND
958
start_search & (tree_depth !=0)
READ
962
access_done
COMPARE
966
string_code,char>prev_code,attach_char
right_child !=0
SEARCH RIGHT
974
right_child==0
left_child==0
string_code,char<prev_code,attach_char
left_child!=0
SEARCH LEFT
970
string_code,char==prev_code,attach_char
FOUND
978

FIG. 16

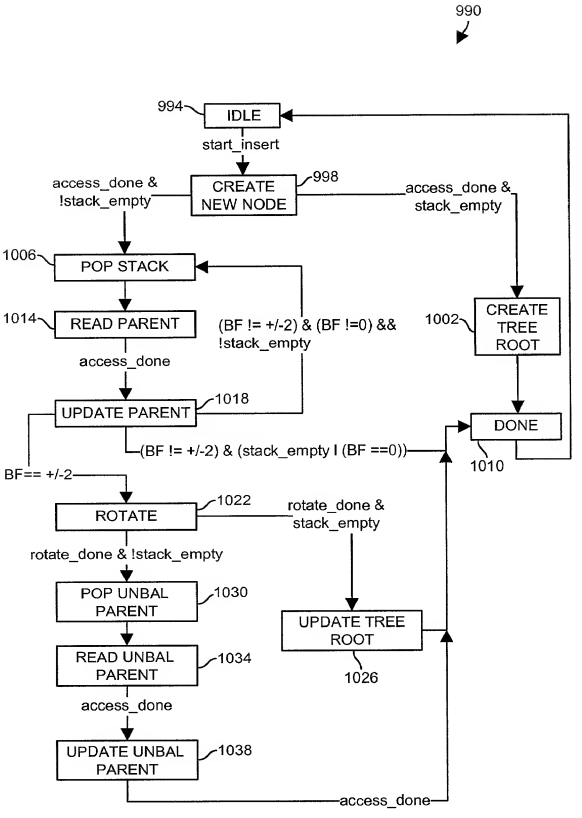

990
994
IDLE
start_insert
access_done & !stack_empty
CREATE NEW NODE
998
access_done & stack_empty
1006
POP STACK
1014
READ PARENT
(BF != +/-2) & (BF !=0) && !stack_empty
1002
CREATE TREE ROOT
access_done
UPDATE PARENT
1018
DONE
(BF != +/-2) & (stack_empty I (BF ==0))
1010
BF== +/-2
ROTATE
1022
rotate_done & stack_empty
rotate_done & !stack_empty
POP UNBAL PARENT
1030
UPDATE TREE ROOT
1026
READ UNBAL PARENT
1034
access_done
UPDATE UNBAL PARENT
1038
access_done

FIG. 17

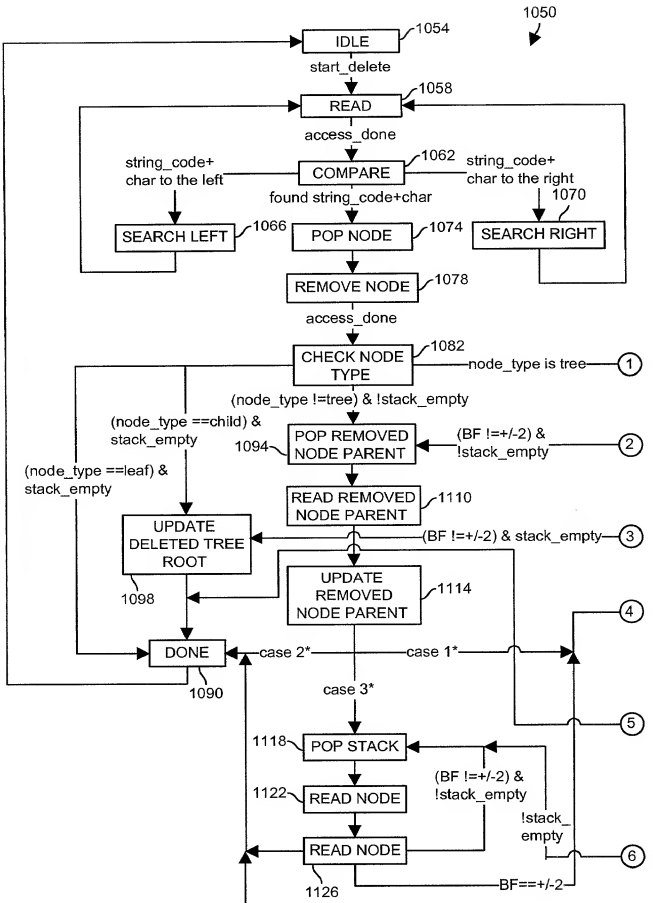
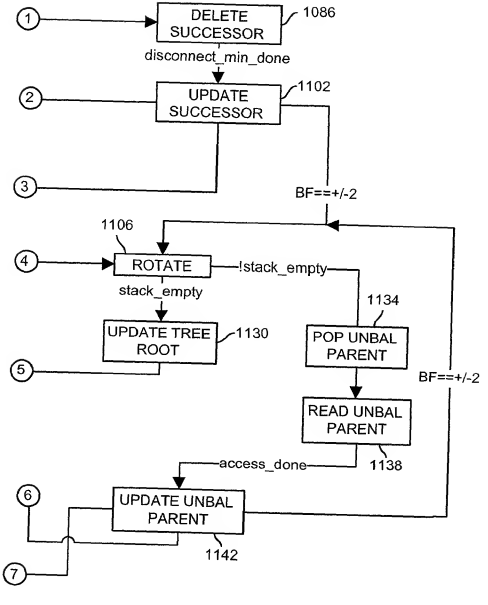

1050
IDLE
1054
start_delete
READ
1058
access_done
string_code+
char to the left
COMPARE
1062
string_code+
char to the right
found string_code+char
1070
SEARCH LEFT
1066
POP NODE
1074
SEARCH RIGHT
REMOVE NODE
1078
access_done
CHECK NODE TYPE
1082
node_type is tree
1
(node_type !=tree) & !stack_empty
(node_type ==child) &
stack_empty
POP REMOVED NODE PARENT
1094
(BF !=+/-2) &
!stack_empty
2
(node_type ==leaf) &
stack_empty
READ REMOVED NODE PARENT
1110
UPDATE DELETED TREE ROOT
(BF !=+/-2) & stack_empty
3
1098
UPDATE REMOVED NODE PARENT
1114
4
DONE
case 2*
case 1*
1090
case 3*
5
1118
POP STACK
(BF !=+/-2) &
!stack_empty
1122
READ NODE
!stack_
empty
READ NODE
6
1126
BF==+/-2
no further height changes
7

FIG. 18A

FIG. 18B


1
DELETE SUCCESSOR
1086
disconnect_min_done
2
UPDATE SUCCESSOR
1102
3
BF==+/-2
1106
4
ROTATE
!stack_empty
stack_empty
1134
UPDATE TREE ROOT
1130
POP UNBAL PARENT
5
BF==+/-2
READ UNBAL PARENT
access_done
1138
6
UPDATE UNBAL PARENT
1142
7

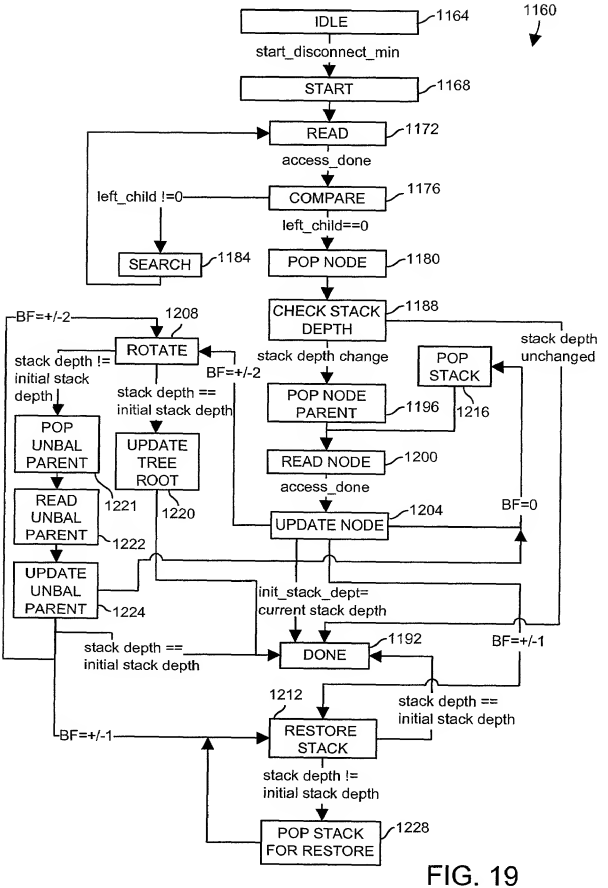

1160
IDLE
1164
start_disconnect_min
START
1168
READ
1172
access_done
left_child !=0
COMPARE
1176
left_child==0
SEARCH
1184
POP NODE
1180
BF=+/-2
1208
CHECK STACK DEPTH
1188
stack depth unchanged
ROTATE
stack depth !=
initial stack
depth
BF=+/-2
stack depth change
POP STACK
1216
stack depth ==
initial stack depth
POP NODE PARENT
1196
POP UNBAL PARENT
UPDATE TREE ROOT
READ NODE
1200
access_done
READ UNBAL PARENT
1221
1220
UPDATE NODE
1204
BF=0
1222
UPDATE UNBAL PARENT
1224
init_stack_dept=
current stack depth
stack depth ==
initial stack depth
DONE
1192
BF=+/-1
1212
stack depth ==
initial stack depth
BF=+/-1
RESTORE STACK
stack depth !=
initial stack depth
POP STACK FOR RESTORE
1228

FIG. 19

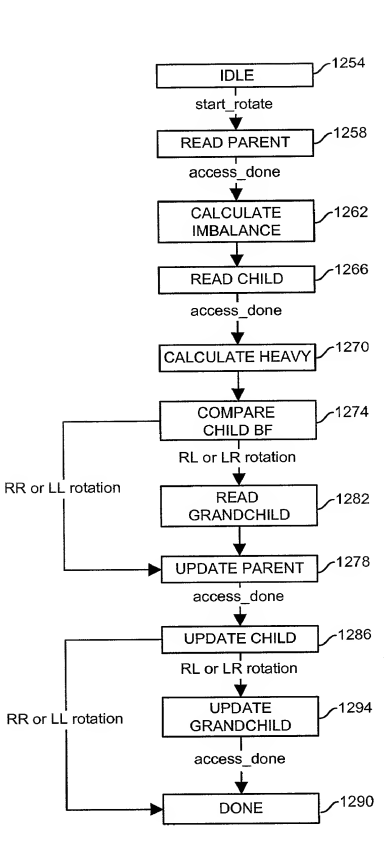

1250
IDLE
1254
start_rotate
READ PARENT
1258
access_done
CALCULATE IMBALANCE
1262
READ CHILD
1266
access_done
CALCULATE HEAVY
1270
COMPARE CHILD BF
1274
RL or LR rotation
RR or LL rotation
READ GRANDCHILD
1282
UPDATE PARENT
1278
access_done
UPDATE CHILD
1286
RL or LR rotation
RR or LL rotation
UPDATE GRANDCHILD
1294
access_done
DONE
1290

FIG. 20

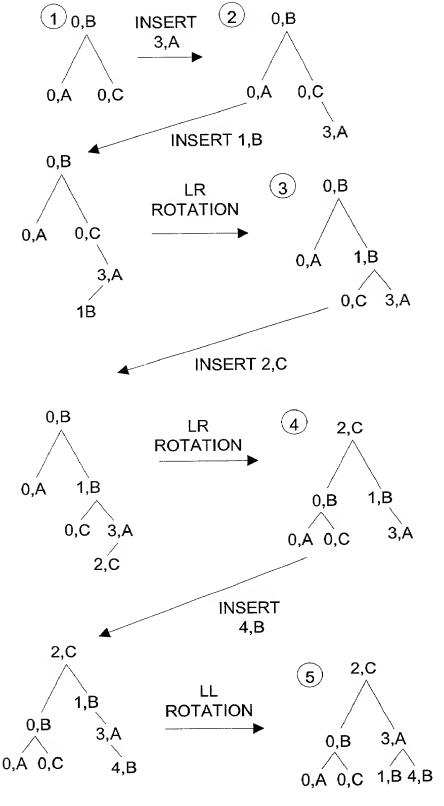

1
0,B
0,A
0,C
INSERT
3,A
2
0,B
0,A
0,C
3,A
INSERT 1,B
0,B
0,A
0,C
3,A
1B
LR
ROTATION
3
0,B
0,A
1,B
0,C
3,A
INSERT 2,C
0,B
0,A
1,B
0,C
3,A
2,C
LR
ROTATION
4
2,C
0,B
1,B
0,A
0,C
3,A
INSERT
4,B
2,C
0,B
1,B
0,A
0,C
3,A
4,B
LL
ROTATION
5
2,C
0,B
3,A
0,A
0,C
1,B
4,B

FIG. 21